L Number	Hits	Search Text	DB	Time stamp
1	6489	(retry$3 or retrie$3) same (cache$1)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:33
2	1658	(retry$3 or retrie$3) same (read$3 or load$4) same (writ$7 or store) same (cache$1)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:51
3	981	(711/$.ccls. or 710/$.ccls.) and ((retry$3 or retrie$3) same (read$3 or load$4) same (writ$7 or store) same (cache$1))	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:56
4	1970	(retry$3 or retrie$3) near10 (read$3 or load$4 or writ$7 or store) same (cache$1)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:40
5	1273	(retry$3 or retrie$3) near10 (read$3 or load$4 or writ$7 or store) near10 (cache$1)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:46
6	108	(retry$3 or retrie$3) near10 (read$3 or load$4 or writ$7 or store) near10 (cache$1 adj1 line$1)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:47
7	88	(711/$.ccls. or 710/$.ccls.) and ((retry$3 or retrie$3) near10 (read$3 or load$4 or writ$7 or store) near10 (cache$1 adj1 line$1))	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:50
8	343	(retry$3 or retries or retried) same (read$3 or load$4) same (writ$7 or store) same (cache$1)	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:52
9	256	(711/$.ccls. or 710/$.ccls.) and ((retry$3 or retries or retried) same (read$3 or load$4) same (writ$7 or store) same (cache$1))	USPAT; EPO; JPO; DERWENT; IBM_TDB	2002/09/07 10:57

L Number	Hits	Search Text	DB	Time stamp
403	5	(read adj1 modif$3 adj1 writ$4) near10 (retry$3 or retries or retried or reissu$5 or (re adj1 issu$4))	USPAT; EPO; JPO; DERWENT; IBM TDB	2002/09/07 13:50

Your search matched **4** of **794619** documents.
Results are shown **15** to a page, sorted by **publication year** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the tex
Then click **Search Again.**

(((retry* or retried or retries or reissu*) and arbit*) and ((1988 <in> py) or (1989 <in> p

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **Blocking in a completely shared resource environment with state d resource and residency requirements**
Kaufman, J.S.
INFOCOM '92. Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, IEEE , 1992
Page(s): 2224 -2232 vol.3

[Abstract] [PDF Full-Text (564 KB)] **CNF**

2 **Concurrent error diagnosis in mesh array architectures based on overlapping H-processes**
Manolakos, E.S.; Dakhil, D.; Vai, M.
Defect and Fault Tolerance on VLSI Systems, 1991. Proceedings., 1991 Inter Workshop on , 1991
Page(s): 139 -152

[Abstract] [PDF Full-Text (480 KB)] **CNF**

3 **Trustworthy bus arbiter by alternate-data retry**
Tokito, K.; Kurokawa, T.; Koga, Y.
Fault Tolerant Systems, 1991. Proceedings., Pacific Rim International Sympo 1991
Page(s): 52 -57

[Abstract] [PDF Full-Text (340 KB)] **CNF**

4 **Bandwidth analysis of multibus parallel subnetworks**
Jung, H.-W.; Saltzman, M.J.
Computers and Communications, 1989. Conference Proceedings., Eighth Ann

International Phoenix Conference on , 1989
Page(s): 391 -394

[Abstract] [PDF Full-Text (252 KB)] **CNF**

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search
Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting
No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Your search matched **[0]** of **[794619]** documents.

You may refine your search by editing the current search expression or entering a new one the text box. Then click search Again.

(((retry* or retried or retries or reissu*) and cach*) and ((1988 <in> py) or (1989 <in> p

Search Again

OR

Use your browser's back button to return to your original search page.

Results:

No documents matched your query.

Your search matched **1** of **794619** documents.
Results are shown **15** to a page, sorted by **publication year** in **descending** order.
You may refine your search by editing the current search expression or entering a new one the tex

Then click **Search Again**.

(((retry* or retried or retries or reissu*) and writ*) and ((1988 <in> py) or (1989 <in> p

Search Again

Results:
Journal or Magazine = **JNL** Conference = **CNF** Standard = **STD**

1 **An analytic treatment of the reliability and performance of mirrored subsystems**
Geist, R.; Trivedi, K.
Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers., The Twenty-Th International Symposium on , Aug. 1993
Page(s): 442 -450

[Abstract] [PDF Full-Text (596 KB)] **CNF**

Search Results

Search Results for: [(retry* or retries or retried or reissu*) and cach*]
Found 432 of 101,097 searched. → Rerun within the Portal

Warning: Maximum result set of 200 exceeded. Consider refining.

Search within Results

GO > Advanced Search > Search Help/Tips

Sort by: Title Publication Publication Date Score

Results 1 - 20 of 200 **short listing**

Prev Page 1 2 3 4 5 6 7 8 9 10 Next Page

1 Modeling a circuit switched multiprocessor interconnect 100%
Daniel Nussbaum , Ingmar Vuong-Adlerberg , Anant Agarwal
ACM SIGMETRICS Performance Evaluation Review , Proceedings of the ACM conference on Measurement and modeling of computer systems April 1990
Volume 18 Issue 1

2 Exception handling and object-oriented programming: towards a synthesis 100%
Christophe Dony
ACM SIGPLAN Notices , Proceedings of the European conference on object-oriented programming on Object-oriented programming systems, languages, and applications September 1990
Volume 25 Issue 10

The paper presents a discussion and a specification of an exception handling system dedicated to object-oriented programming. We show how a full object-oriented representation of exceptions and of protocols to handle them, using meta-classes, makes the system powerful as well as

extendible and solves many classical exception handling issues. We explain the interest for object-oriented programming of handlers attached to classes and to expressions. We propose an original algorithm for propag ...

3 Cache coherence for large scale shared memory multiprocessors 100%

M. Thapar , B. Delagi

Proceedings of the second annual ACM symposium on Parallel algorithms and architectures May 1990

4 The LEAP load and test driver 100%

T. A. Dolotta , J. S. Licwinko , R. E. Menninger , W. D. Roome

Proceedings of the 2nd international conference on Software engineering October 1976

LEAP is a facility for testing interactive applications that are implemented on IBM System/360 and System/370 computers; it simulates concurrently the actions of several IBM 3270-type terminals and of their operators. LEAP allows one to organize, in a systematic and efficient fashion, a large-scale, repeatable testing program for such an interactive application system. LEAP is one of the tools of the Programmer's Work-bench. It operates under the UNIX Time Sharing System; all test preparati ...

5 An open operating system for a single-user machine 100%

Butler W. Lampson , Robert F. Sproull

Proceedings of the seventh symposium on Operating systems principles December 1979

The file system and modularization of a single-user operating system are described. The main points of interest are the openness of the system, which establishes no sharp boundary between itself and the user's programs, and the techniques used to make the system robust.

6 Process backup in producer-consumer systems 100%

David L. Russell

Proceedings of the sixth symposium on Operating systems principles November 1977

System state restoration after detection of an error is discussed for producer-consumer systems, with emphasis on the control of the domino effect. Recovery primitives MARK, RESTORE, and PURGE are proposed that, in conjunction with the use of SEND-RECEIVE interprocess communication primitives, allow bounds to be placed on the amount of unnecessary restoration

that can occur as a result of system state restoration.

7 Hardware monitoring of real-time aerospace computer systems 100%
D. R. Partridge , R. E. Card
Proceedings of the international symposium on Computer performance modeling measurement and evaluation March 1976

Hardware monitoring has proven to be a useful means for measuring the performance of computer systems generally, and is particularly attractive for use on real-time systems due to its attribute of non-interference with system operation. This technique is uniquely able to quantify precisely the interactions between hardware and software, which must be completely understood in these systems. In this paper, we report the application of a commercially-developed hardware monitor to two real-time ...

8 Computer structures: What have we learned from the PDP-11? 100%
Gordon Bell , William D. Strecker
Proceedings of the third annual symposium on Computer architecture January 1976

Over the PDP-11'S six year life about 20,000 specimens have been built based on 10 species (models). Although range was a design goal, it was unquantified; the actual range has exceeded expectations (500:1 in memory size and system price). The range has stressed the basic mini(mal) computer architecture along all dimensions. The main PMS structure, i.e. the UNIBUS, has been adopted as a de facto standard of interconnection for many micro and minicomputer systems. The architectural experienc ...

9 The NYU Ultracomputer—designing a MIMD, shared-memory parallel machine (Extended Abstract) 100%
Allan Gottlieb , Ralph Grishman , Clyde P. Kruskal , Kevin P. McAuliffe , Larry Rudolph , Marc Snir
Proceedings of the ninth annual symposium on Computer Architecture April 1982

We present the design for the NYU Ultracomputer, a shared-memory MIMD parallel machine composed of thousands of autonomous processing elements. This machine uses an enhanced message switching network with the geometry of an Omega-network to approximate the ideal behavior of Schwartz's paracomputer model of computation and to implement efficiently the important fetch-and-add synchronization primitive. We outline the hardware that would be required to build a 4096 processor system using 1990' ...

10 Incorporation of multiaccess links into a routing protocol 100%

Radia Perlman

Proceedings of the 8th Data Communications Symposium October 1983

Conventional routing protocols and algorithms work most efficiently on sparsely connected networks. Network topologies today include multiaccess links which include hundreds of nodes, all of which are capable of direct communication with each other. Some of these multiaccess links have broadcasting ability. This paper presents protocols and algorithms for efficiently dealing with the characteristics of various types of multiaccess links, when they are included as links in a general topology ...

11 A multiprocessor implementation of relaxation-based electrical circuit simulation 100%

J. T. Deutsch , A. R. Newton

21st Proceedings of the Design Automation Conference on Design automation June 1984

The electrical circuit simulation of large integrated circuits is very expensive. New relaxation-based algorithms promise to reduce this cost by exploiting the properties of large networks. However, this speed improvement is not sufficient for the cost-effective analysis of very large circuits. While array processors have helped inprove the performance of circuit simulators, further improvement can be achieved by the use of special-purpose multiprocessors. In this paper, the implementation ...

12 Combining produce and consume operations in a pipelined shared memory multiprocessor 100%

B. J. Rodriquez , H. F. Jordan

Proceedings of the 1989 conference on Supercomputing August 1989

A technique for combining produce and consume operations is proposed. An Omega network was designed to work with a pipelined micro multiprocessor system, and each of its nodes to be implemented on a single chip. Without a significant increase in hardware complexity, the nodes combine produce and consume messages in an unbounded way. The combining technique that was developed guarantees that every produce and consume message that is combined will always succeed, adding a new benefit to combi ...

13 The performance implications of thread management alternatives for shared-memory multiprocessors 100%

T. E. Anderson , D. D. Lazowska , H. M. Levy

ACM SIGMETRICS Performance Evaluation Review , Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems April 1989

Volume 17 Issue 1

Threads (“lightweight” processes) have become a common element of new languages and operating systems. This paper examines the performance implications of several data structure and algorithm alternatives for thread management in shared-memory multiprocessors. Both experimental measurements and analytical model projections are presented. For applications with fine-grained parallelism, small differences in thread management are shown to have significant performance imp ...

14 Adaptive backoff synchronization techniques 100%

A. Agarwal , M. Cherian

ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture April 1989

Volume 17 Issue 3

Shared-memory multiprocessors commonly use shared variables for synchronization. Our simulations of real parallel applications show that large-scale cache-coherent multiprocessors suffer significant amounts of invalidation traffic due to synchronization. Large multiprocessors that do not cache synchronization variables are often more severely impacted. If this synchronization traffic is not reduced or managed adequately, synchronization references can cause severe congestion in the network. ...

15 Evaluation of memory system for integrated Prolog processor IPP 100%

M. Morioka , S. Yamaguchi , T. Bandoh

ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture April 1989

Volume 17 Issue 3

This paper discusses an optimal memory system to realize a high performance integrated Prolog processor, the IPP. First, the memory access characteristics of Prolog are analyzed by a simulator, which simulates the execution of a Prolog program at a micro instruction level. The main findings from this analysis are that: the write access ratio of Prolog is larger than that of

procedural languages; and performance improvement requires the memory system to process concentrated, large write acce ...

16 A high performance Prolog processor with multiple function units 100%

A. Singhal , Y. N. Patt

ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture April 1989

Volume 17 Issue 3

We describe the Parallel Unification Machine (PLUM), a Prolog processor that exploits fine grain parallelism using multiple function units executing in parallel. In most cases the execution of bookkeeping instructions is almost completely overlapped by unification, and the performance of the processor is limited only by the available unification parallelism. We present measurements from a register transfer level simulator of PLUM. These results show that PLUM with 3 Unification Units achiev ...

17 Mirage: a coherent distributed shared memory design 100%

B. Fleisch , G. Popek

ACM SIGOPS Operating Systems Review , Proceedings of the Twelfth ACM symposium on Operating systems principles November 1989

Volume 23 Issue 5

Shared memory is an effective and efficient paradigm for interprocess communication. We are concerned with software that makes use of shared memory in a single site system and its extension to a multimachine environment. Here we describe the design of a distributed shared memory (DSM) system called Mirage developed at UCLA. Mirage provides a form of network transparency to make network boundaries invisible for shared memory and is upward compatible with an existing interfac ...

18 Threads and input/output in the synthesis kernal 100%

H. Massalin , C. Pu

ACM SIGOPS Operating Systems Review , Proceedings of the Twelfth ACM symposium on Operating systems principles November 1989

Volume 23 Issue 5

The Synthesis operating system kernel combines several techniques to provide high performance, including kernel code synthesis, fine-grain scheduling, and optimistic synchronization. Kernel code synthesis reduces the execution path for frequently used kernel calls. Optimistic synchronization increases

concurrency within the kernel. Their combination results in significant performance improvement over traditional operating system implementations. Using hardware and software emulating a SUN 3 ...

19 The design of nectar: a network backplane for heterogeneous multicomputers 100%

Emmanuel Arnould , H. T. Kung , Francois Bitz , Robert D. Sansom , Eric C. Cooperm

ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems April 1989 Volume 17 Issue 2

Nectar is a “network backplane” for use in heterogeneous multicomputers. The initial system consists of a star-shaped fiber-optic network with an aggregate bandwidth of 1.6 gigabits/second and a switching latency of 700 nanoseconds. The system can be scaled up by connecting hundreds of these networks together. The Nectar architecture provides a flexible way to handle heterogeneity and task-level parallelism. A wide variety of machines can be connected as Nectar nodes ...

20 A unified vector/scalar floating-point architecture 100%

N. P. Jouppi , J. Bertoni , D. W. Wall

ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems April 1989 Volume 17 Issue 2

In this paper we present a unified approach to vector and scalar computation, using a single register file for both scalar operands and vector elements. The goal of this architecture is to yield improved scalar performance while broadening the range of vectorizable applications. For example, reduction operations and recurrences can be expressed in vector form in this architecture. This approach results in greater overall performance for most applications than does the approach of emphasizin ...